A Novel Soap Recycling Embedded System for Greener Hospitality

A staggering 440 *billion* pounds of solid waste – equivalent to the weight of 2 million blue whales, is generated by hotels every year[1]. Shockingly, over 5 million partially-used soap bars end up in landfills *every day* in the *US alone*[2]. This not only causes economic loss and waste of natural resources, but the hospitality industry also inadvertently contributes to environmental crises such as landfill pollution, water contamination, and greenhouse gas emissions. Unfortunately, there is no practical solution to tackle this serious global problem; and so I was inspired to solve this issue. My unique solution stems from my personal experience as my family owns hotels before I was born.

Currently, the only existing solution is to collect the leftover soaps from all the hotels (700,000 hotels worldwide[3] – so, not practical), then melt and mold them to make new soaps. But, it's not very functional – involves huge logistical costs, requires many big factories globally, has only some participating hotels, and so doesn't have a global reach. But, my innovative approach stands out. I created a practical, eco-friendly, and cheap novel device that recycles the soap waste at the source (hotels) itself, eliminating all logistics. I used soap's natural tendency to break down in water, so I also eliminated the cost of heating soaps for molding. My revolutionary device can seamlessly integrate with hotels' existing laundry systems.

My fully-tested and functional prototype is simple to use and doesn't require any training, but at the core lies a sophisticated mechanism. My device integrates two microcontrollers to execute two programs simultaneously (see attached project photos): 1. auto mixing (container A), 2. auto dispensing (container B). Instead of throwing away the leftover soaps, housekeepers can just toss it in container A, which is connected to a tap for warm water and to a motor (SG90 servo motor coded using ESP32Servo.h library and driven by ESP32 microcontroller) for mixing (in both directions) because I conducted tests and found that soap dissolves faster in warm water with some stirring. This forms a soapy solution that is added to the washer automatically, along with the hotel's regular laundry detergent. Also, I coded it such that the mixing duration is customizable – depending on the size and needs of hotels. When the user initiates the push button, the Arduino Uno interacts with various components and *only when* container B is detected empty or partially-full (or otherwise it alerts that it is full), the stepper motor engages, automatically turning the tube downwards to initiate soapy solution flow from container A to B until a rain level sensor in container B detects that the level reaches the predefined limit. Then, the motor retracts the tube and also alerts the user with a piezo buzzer sound and the blue LED lighting up. Finally, the soapy solution goes from container B to the laundry machine via tube, with the hotel's existing laundry system. I used a 28BYJ-48 Stepper Motor with ULN2003 Driver. For precise stepper motor control, the Stepper.h library was employed, utilizing four digital output pins. Two stepper objects were created in code to manage motor direction. The rain level sensor, an input device, was positioned at a specific height within container B to detect a predetermined amount – it is also customizable according to the hotel's size and needs.

My recycled soapy solution has many other applications as well – can be used to clean washrooms (by adding vinegar to it), and also as an insecticide and weed control. My prototype is tested and fully functional with 100% accuracy. My unique device completely eliminates soap and shampoo wastage by recycling it, so it also protects our scarce natural resources. My device also saves hotels – money, as it reduces their cost of detergent and waste management; and it also saves the environment from landfill pollution and water contamination. Therefore, my novel soap recycling device will be a game-changer for the hospitality industry and the environment.

Works Cited

1. Crockett, Zachary. “The Surprising Afterlife of Used Hotel Soap.” *The Hustle*, 22 Apr. 2022, thehustle.co/the-surprising-afterlife-of-used-hotel-soap/.
2. “‘Wasted’: Have You Ever Wondered What Happens to Your Half-Used Bar of Hotel Soap? | CBC Radio.” *CBCnews*, CBC/Radio Canada, 10 Feb. 2018, www.cbc.ca/radio/day6/episode-376-boushie-verdict-human-rights-in-the-philippines-ads-in-space-releasethememo-and-more-1.4523359/wasted-have-you-ever-wondered-what-happens-to-your-half-used-bar-of-hotel-soap-1.4523387.
3. “Hotel Industry Statistics 2023.” *Gay Travel*, 24 July 2023, www.gaytravel.com/gay-blog/hotel-industry-statistics.

GitHub Repository: Hotel Soap Recycling Project

Code for this project:

mixer.ino:

```
#include <ESP32Servo.h>
Servo myservo;
const int custom = 100000; //enter custom duration in ms
const int servoPin = 2;
int cond = 1;
void setup() {
  myservo.setPeriodHertz(50);
  myservo.attach(servoPin);
}
void loop() {
 while(cond){
   for(int pos = 0; pos < 360; pos += 1) {
 myservo.write(pos);
 delay(15);
 }
 }
 delay(custom);
cond = 0;
}
```

soap_getter.ino:

```
#include <Stepper.h>
const int stepsPerRevolution = 523, steps = 175;
Stepper myStepper(stepsPerRevolution, 8, 10, 9, 11);// counter clockwise Stepper object
```

```
Stepper myStepper2(stepsPerRevolution, 8, 9, 10, 11);// clockwise Stepper object
#define power 6
#define rainLevelSensor 7
#define button 5
#define LED_Buzzer 4
void setup() {
  pinMode(power, OUTPUT);
  pinMode(rainLevelSensor, INPUT);
  digitalWrite(power, HIGH);
  myStepper.setSpeed(60);
  myStepper2.setSpeed(20);
  pinMode(button, INPUT_PULLUP);
  pinMode(LED_Buzzer, OUTPUT);
}
void loop(){
  int but = digitalRead(button);
  int val = digitalRead(rainLevelSensor);
  if(but == LOW){
  if(val){
  myStepper.step(stepsPerRevolution);
  delay(5);
  while(val){
  val = digitalRead(rainLevelSensor);
  delay(5);
  }
  for(int i = 0; i < 3; i++){
  myStepper2.step(steps);
  delay(5);
  }
  tone(LED_Buzzer, 1000);
  delay(3000);
  noTone(LED_Buzzer);
  }
  else{
  tone(LED_Buzzer, 1000);
  delay(3000);
  noTone(LED_Buzzer);
  }
  }
}
```